REMARKS

It is respectfully submitted that the Application is now allowable. Accordingly, applicant respectfully requests reconsideration and allowance of the application. The Commissioner is authorized to charge any fees, including but not limited to extension of time and additional claims fees, due but not submitted with this paper to Deposit Account No. 07-0153. The Examiner is respectfully requested to call applicant's Attorney if or any reasons that would advance the current application to issue. Please reference Attorney Docket No. 131195-1003.

Dated: _____

Respectfully submitted,
GARDERE WYNNE SEWELL LLP

_____
Marc A. Hubbard
Registration No. 32,506
ATTORNEY FOR APPLICANT

1601 Elm Street, Suite 3000
Dallas, Texas 75201-4761
(214) 999-4880 - Telephone
(214) 999-3880 - Facsimile

The *New* McGraw-Hill Telecom Factbook

Second Edition

Joseph A. Pecar
David A. Garbin

McGraw-Hill
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

[redacted]

Library of Congress Cataloging-in-Publication Data

Pecar, Joseph A.
The new McGraw-Hill telecom factbook / Joe Pecar, David Garbin.
p. cm.
ISBN 0-07-135163-9
1. Telecommunication. I. Garbin, David A. II. Title

TK5101 P334 2000
384—dc21

00-055011

McGraw-Hill
A Division of The McGraw-Hill Companies

234567890 QM/QM 0987654321

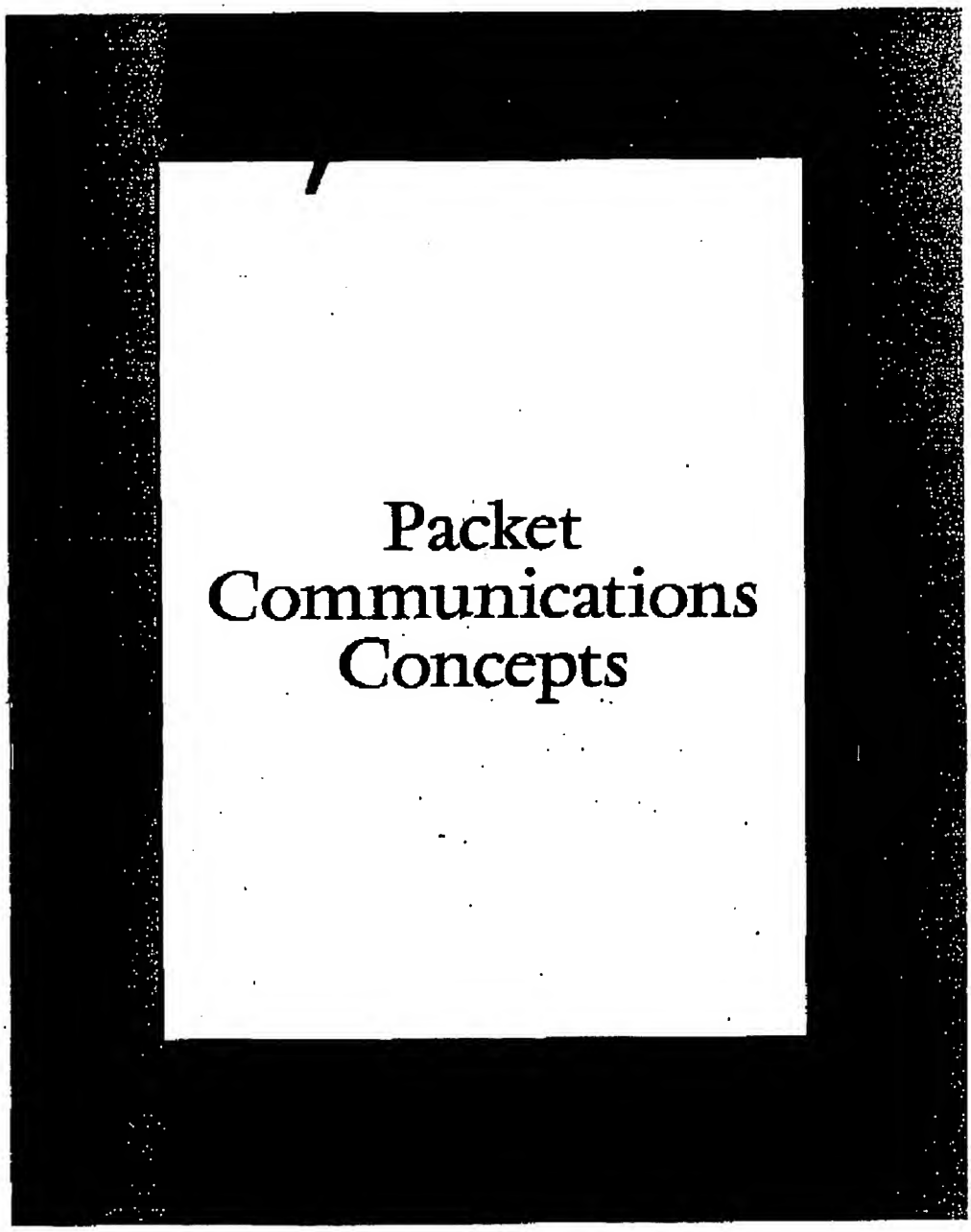
ISBN 0-07-135163-9

The sponsoring editor for this book was Steve Chapman and the production manager was Pamela Pelton. It was set in Vendome by Patricia Wallenburg.

Printed and bound by Quebecor/Martinsburg.

This book is printed on recycled, acid-free paper containing a minimum of 50% recycled, de-inked fiber.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to the Director of Special Sales, McGraw-Hill, Two Penn Plaza, New York, NY 10121-2298. Or contact your local bookstore.


Packet
Communications
Concepts

This chapter presents fundamentals of data communications in preparation for a discussion of data services in forthcoming chapters. It relies on the introductory material presented earlier for terms of reference and background. In particular, it builds on the definitions and descriptions of digital electrical signals, binary bits, error detection and correction, data terminal equipment (DTE), digital carrier, time division multiplexing (TDM), and digital circuit switching presented in Chapters 1 through 6.

A significant difference between voice and data service is the extent to which human intervention is required to ensure end-to-end communications integrity, including diagnosis and recovery under failed or inadequate service conditions. For example, if an American places a telephone call to Japan that is answered by someone who cannot speak English, human intelligence is relied upon to seek an interpreter or to take alternative action. Similarly, if a call cannot be completed due to a network failure, a human determines the problem and takes corrective steps.

By contrast, data services are provided with minimal human intervention. As a consequence, more elaborate mechanisms are required to ensure that transmitting and receiving DTEs "speak the same language," and that service restoration actions are promptly taken under network failure conditions. This generally requires higher levels of hardware and software compatibility among DTEs and intervening data network elements than is required in voice networks.

For private data networks, it might be feasible to specify hardware and software from a single source, achieving compatibility through proprietary design. For public networks relying on universal connectivity supported by multiple vendors, standards and protocols defined by U.S. and worldwide organizations must be used. *Protocols* are strict procedures for the initiation, maintenance and termination of data communications, as described later in this chapter.

As we saw in Chapter 1, traffic characteristics impose different requirements on voice-versus-data network design. For circuit switched voice communications, a nominal post-dial delay (call setup) interval of several seconds is acceptable. However, data traffic often occurs in short bursts, resulting in long inactive periods interspersed with high-speed information exchange. So a dedicated non-switched channel would result in inefficient network utilization. In addition, setup time to establish a circuit-switched call would result in unacceptable response times for on-line data transactions, where terminal-operator requests for data must be responded to in a very few seconds.

Developed in the 1970s for long-distance data communications, packet switching, an alternative to circuit switching, drastically reduces or eliminates call setup time and inactive periods on circuits and is therefore well suited to bursty data traffic.

This chapter introduces basic packet switching principles and fundamental concepts underlying all protocols. It shows how different sets of protocols evolved as new technology changed the constraints under which the protocols operated. Finally, it describes the operation of major local and wide area network services in use today in terms of facility types and protocols.

Packet Switching Fundamentals

A *packet* is a quantity of data that is transmitted and switched as a composite whole. A packet contains user data, destination and source information, control information, and error-detection bits, arranged in a particular format. A typical packet is shown in Figure 7.1. Packets are formed by segmenting user message information or data (which may be any number of bits or bytes) into packets of limited length by *packet assembler-disassemblers* (PADs), as shown in the figure. Packetization is used in virtually all data communications systems.

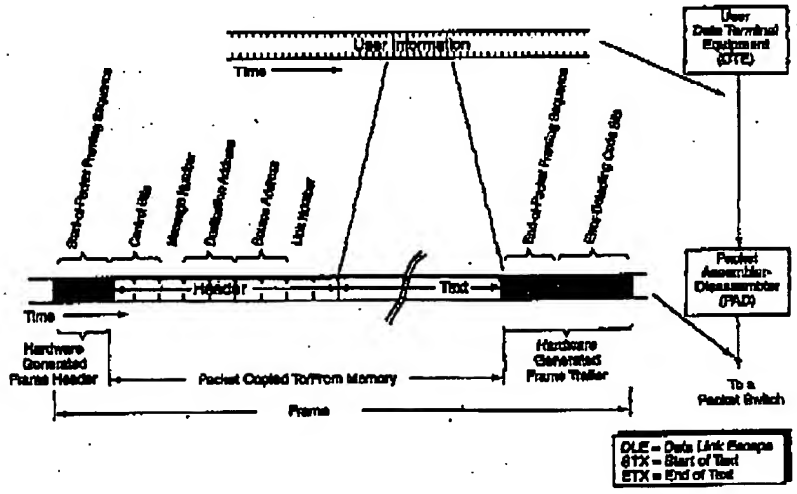


Figure 7.1 Typical packet format.

Sufficient information is embedded in packets to enable packet switches to route them through networks. A *packet header*, which precedes user data, may contain destination address, source address, link numbers, packet numbers, and other information. Specifically, a header is control information appended to a segment of user data for synchronization, routing, and sequencing of a transmitted data packet. Among adjacent and connected switching nodes, packets are encapsulated in *frames* which themselves include headers and trailers (codes), usually hardware generated, to indicate start-of-message and end-of-message events. The glossary explains in more detail several legitimate meanings for the word "frame" when used in telecommunications contexts.

A packet-switching network is designed to switch and transport information in packet form. Figure 7.2 illustrates how packet switching works and the differences between packet and circuit switching. In the figure, user messages, represented by the rectangles labeled A, B, and C are shown as DTE inputs and outputs. Message length is indicated by the length of the rectangles.

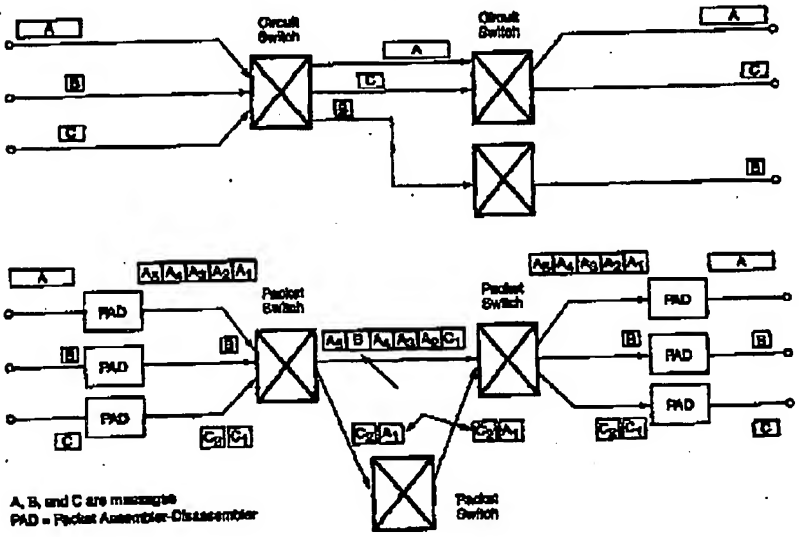


Figure 7.2
Example of circuit and packet-switched connections.

For circuit switching, illustrated in the upper half of Figure 7.2, channels between switches are used exclusively for individual message transmissions A, B, and C, assigned on a first-come, first-served basis. The circuit switches establish connections (data calls) between remote DTEs in a manner similar to that for voice traffic between two telephones. As with voice, channels remain occupied (out of service for additional calls) until released. As already noted, for interactive message traffic generated by human keyboard operators, actual information transfer may only occur in bursts, interspersed with long periods of inactivity. In this case, circuit switching makes inefficient use of potentially expensive transmission resources. Once all links are busy, new messages, even short ones, may experience unacceptable delay waiting for circuits to clear. In this example, three circuits are required between the switches to handle the information transfer.

In packet switching, messages A, B, and C are segmented into packets by PADs prior to being offered to packet switches. This operation is illustrated in the lower half of Figure 7.2 where packets corresponding to messages A, B, and C are processed by packet switches and interleaved on transmission links between the switches. Packet-switched networks provide more efficient data transport than circuit-switched networks because the connections through the network are used only while data is being transmitted. As a result, many different connections can share the same circuit.

Each packet switch is connected to one or more remote packet switches. In the transmission of message A, note in the figure that some packets from message A are delayed more than others. This occurs when packets from message A wait in a queue at the switch while packets from other messages are transmitted over the same circuit. Because of the random nature of packet arrivals from different sources, this situation occurs during normal operations. Packet switches must provide storage space for packets that are waiting, in the form of buffers. Because of this phenomenon, packet switching operations produce variable end-to-end message delays. For data applications this is normally not a problem, but it can degrade voice and video communications. For this and other reasons, packet switching has traditionally been used exclusively for data communications. Advanced, fast packet-switching technologies supporting voice, data, video, and other services overcome these limitations.

Packet-Switch Functions and Capabilities

A packet switch consists of the following functional entities:

- Input and output buffering (memory elements to temporarily store packets).
- Processing for decoding header address, routing, and other information; error detection; and switch and network control.
- Internal switching to connect input and output buffers. The transmission of packets through a network requires three types of packet control procedures:
  - Routing control to determine the routes over which packets are transmitted.
  - Flow control to prevent congestion in the network and lock-ups or traffic jams.
  - Error control to deal with any transmission errors that occur.

In contrast to circuit-switched voice networks where signaling is invoked once to establish call connections for the duration of the entire call or transaction, in packet networks, each packet is examined for source and destination address information and acted upon accordingly. While this operation does result in efficient utilization of transmission resources, overall routing, error, and flow control impose significant processing requirements on packet switches. In fact, the throughput of packet networks is limited primarily by the processing capabilities of the packet switches.

In the Internet backbones of the major Network Service Providers (NSPs), the switches have a capacity exceeding 10 million packets per second. This dramatic improvement to the 300-packets-per-second performance of the switches in the 1970s networks has made possible the public data networks we depend upon today.

Access and Transport Services

As noted above, once a call is established in circuit-switched networks, a dedicated, physical connection is established between telephones or other user station equipment, to be torn down at the end of the call.

Analogously, a connection-oriented packet-switched network transport service establishes logical connections in response to station equipment (DTE) requests. All packets entering the networks are delivered to terminating DTEs in the order in which they were received. As with voice calls, *connection-oriented* data services use separate procedures for connection establishment and end-to-end information transfer (connection establishment must take place prior to information transfer).

This service is referred to as *virtual circuit service* since, in the absence of degradation, message routing is logically identical to routing over circuit-switched facilities (i.e., all packets for a given logical connection follow an identical path through the network). Note, however, that circuit-switching inefficiencies are avoided since packets from multiple sources can be interleaved over the same physical transmission paths.

A *permanent virtual circuit* (PVC) is a virtual circuit resembling a leased line in that invariant logical numbers identifying PVCs are dedicated to a single user. Thus, at a particular interface point a network service provider assigns a fixed number of virtual circuits to a user, each of which connects specific network/user interface points. Alternatively, a *switched virtual circuit* (SVC) permits a user to establish virtual circuits between arbitrary network interface points, much like direct-distance dialing in circuit-switched voice networks.

Although most packet-switched networks used for wide area or long-distance data communications offer users virtual circuit service, networks can be designed to offer users *connectionless* service where economics dictate simple switches and control procedures. It eliminates connection set-up, lowers overhead, and results in faster transmission times. Packets are routed independently over the network from source to destination and are delivered in whatever order they arrive at the destination. Connectionless modes are widely used in local area networks to reduce complexity and cost.

Figure 7.3 shows two methods for physically accessing packet switched networks. On the right-hand side of the figure is a host computer (i.e., any computer running a full protocol stack up to the Application layer) attached to a communications *front-end processor* (FEP) with integral PAD functional capabilities. The FEP is connected directly to a packet switch, which is either located on a customer's premises, or connected via digital access facilities.

FEPs, also called stored program communications controllers, are dedicated computers or systems of computers that control data communications between host processors and various types of data communications networks. FEP functions include route selection, multi-

host access, data switching, network management, message sequencing, and flow control. PEPs support both private and public data network operations. For example, the IBM FEPs support private IBM System Network Architecture (SNA) networks, but with network packet-switching interface programs, they can connect with public packet switched networks.

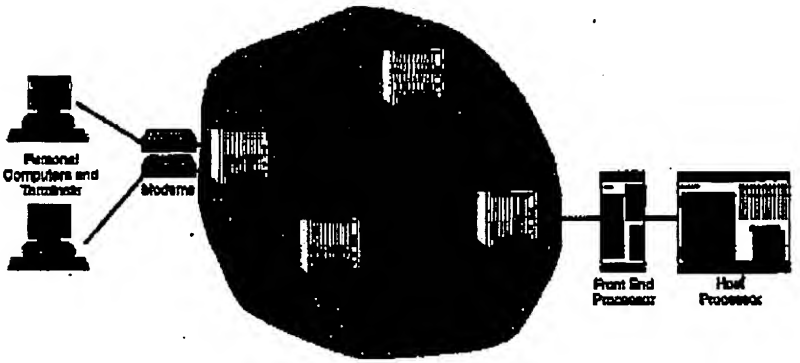


Figure 7.3
Access to packet-switched networks.

The left side of the figure shows how modems are used to connect terminals or other DTEs to remote packet switches and PADs. In this case either dedicated (leased) or public-switched dial-up voice network services can be used to access packet-switched network services using modems. With dial-up service, log-on to the packet network is required each time a user wishes to obtain service.

For the special case of the Internet, the packet switches and the FEPs are implemented as routers. *Routers* are described in the Layer 3 Network part of the "ISO Reference Model for OSI" subsection below. Router functions and operations are described later in this section and throughout the remainder of the book.

Protocol Fundamentals

From the foregoing, it is evident that data communications networks require a high degree of compatibility and interoperability among DTEs and network elements, particularly with respect to physical and

logical interfaces and controls. A challenge is presented by different vendor equipment and/or even different models from the same vendor, all of which must be interconnected.

In 1977, the International Organization for Standardization (ISO) established a subcommittee to develop a standards architecture to achieve the long-term goal of open systems interconnection (OSI). ISO is a voluntary international body concerned with developing standards for a variety of subjects. Data communications standards are developed through the workings of its Technical Committee 97. ISO membership is mainly composed of national standards-making organizations, for example, the American National Standards Institute (ANSI) in the United States, as discussed in Appendix A.

The term open systems interconnection denotes standards for the exchange of information among systems that are *"open"* to one another by virtue of incorporating ISO or other industry accepted standards. The fact that a system is open does not imply any particular system's implementation, technology, or means of interconnection but refers to compliance with applicable standards.

ISO has specified an *OSI Reference Model* that segments communications functions into seven layers. Each layer is assigned related subsets of communications functions implemented in a DTE required to communicate with another DTE. Each layer relies on the next lower layer to perform more primitive functions, and in turn provides services to support the next higher layer. Layers are defined so that changes in one layer do not affect other layers.

Information exchange occurs when corresponding (peer) layers in two systems communicate by means of a set of rules known as protocols. Protocols define the *syntax* (arrangements, formats, and patterns of bits and bytes) and the *semantics* (system control, information context or meaning of patterns of bits or bytes) of exchanged data, as well as numerous other characteristics such as data rates, timing, etc.

Defining the details of seven layers of protocols for data communications is an enormously complex task. Before delving into a technical discussion of the ISO layers, consider an example taken from a more time-honored form of communications. Figure 7.4 illustrates multiple layers of communications between two diplomats from different countries. The exchange of ideas between the two diplomats represents ISO Layer 7, user-to-user communications.

Since the diplomats have no common language, they each engage the services of a translator. The translator converts the message into a common language (e.g., French), writes it down on paper, places the

letter in an addressed envelope, and mails the letter. The envelope is carried in sacks by trains between post offices in the originator's country. At each intermediate post office, the envelope is retrieved from the sack, the destination address is read, and the envelope is placed in a new sack on a different train to continue on its way to the destination. The process of receiving the envelope at each post office, verifying that it is in good condition, and passing it on to be routed to its destination represents ISO Layer 2, Link-Layer communications (the trains themselves are Layer 1, Physical-Layer communications). If Layer 2 is connection oriented, a message is sent to the post office at the origin end of the link noting the condition of the envelope. If the envelope was damaged, a new copy is sent on the next train.

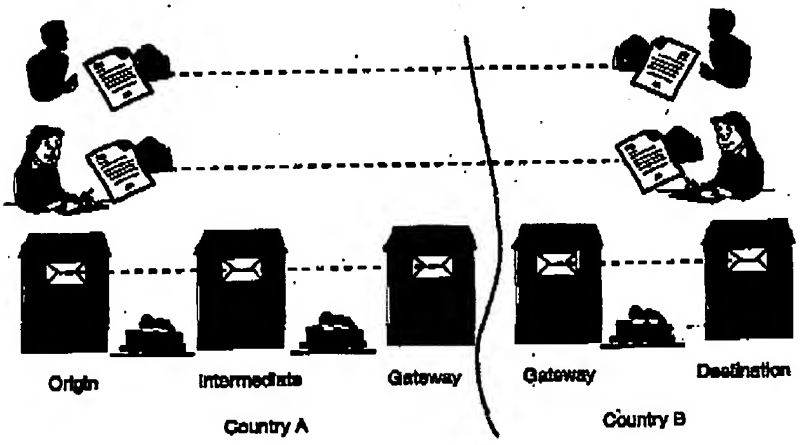


Figure 7.4 Example of multilayer peer-to-peer communications.

The reading of the destination address and the routing of the letter is one of the functions of ISO Layer 3, the Network Layer. The other function of Layer 3 occurs when the letter reaches its final post office (in this case, the gateway post office for the national postal system). In connection-oriented networks, the letter is verified to be in good condition and a message is sent back to the originating post office telling it that the letter arrived at the network boundary successfully. If the letter is damaged, this information is passed back to the originating post office, which will then resend a new copy of the letter. In connectionless networks, letters are passed along between networks without regard to their condition; other means are employed to notify the originator of problems with the message.

At the gateway boundary between the two countries, the letter is passed to the national postal network of the destination country using an agreed-upon process known as an internetwork protocol, a special case of Layer 3. The process of passing the letter from the gateway post office to the post office serving the second diplomat uses the Layer 3 and Layer 2 protocols of the destination country. Once the letter is delivered to the translator, he checks it for integrity. This layer of communications between translators is representative of ISO Layer 4, the Transport Layer. This layer is especially important since it deals with the end-to-end quality and reliability of the communications path between the users. Like the Network Layer, the Transport Layer may be connection-oriented or connectionless. A connection-oriented Transport Layer notifies the originating translator that the letter was received intact. If it was not, the originating translator (not the originating post office) resends a new copy of the letter. Once the letter's integrity is verified by the translator, it is passed to his diplomat for reading.

Note that the translators could change to English to write to each other without affecting either the Layer 1, 2, or 3 processes. Similarly, neither message integrity at Layer 3 nor the translation process of Layer 4 is affected should the physical transportation media change from trains to trucks.

Tradeoffs in Protocol Design

At this point, it is worth stepping back and considering the implications of what we have just learned from the above example. We have seen that connection-oriented transmission ensures the successful receipt of the message between any two end-points where it is used. One might assume, then, that it would be used at every protocol layer in the system. This would be the case if there were not a considerable penalty to be paid in complexity, cost, and performance for using connection-oriented techniques.

In the above example, connection-oriented transmission is available at Layers 2, 3, and 4. At Layer 2, each post office must store copies of each letter sent on the outgoing trains and keep these copies until a message (known as an *acknowledgment*) arrives on a returning train that the letters were received correctly at the next station. At Layer 3, additional copies of each letter are stored at originating post offices

waiting for return messages from destination post offices serving the recipient of the letter or serving a network boundary. Again, the letters are stored until the acknowledgments are received.

The process happens again at Layer 4, with the translator storing copies until he hears from the destination translator. In most cases, the sender will not send more letters to the same destination until he has positive acknowledgment of those already sent. The delays encountered in waiting for acknowledgment messages and the cost of storage are significant considerations for the use of connection-oriented protocols. When and where, then, are these techniques employed? If reliable transmission is required at Layer 4, and it usually is required, then it makes sense always to employ connection-oriented transport protocols. Given that, are these techniques required at the lower layers?

In the early days of digital transmission, error rates on links were relatively high. Since many links were required to complete a path across a network, connection-oriented protocols at Layer 2 were a necessity. Without them, the Layer 3 and above protocols would constantly be detecting errors and asking for retransmissions. The network would be clogged with the acknowledgment messages and the retransmissions, which would also contain errors.

Today, modern digital transmission systems are virtually free of errors and connection-oriented protocols are rarely used at Layer 2. Some systems still retain the connection-oriented paradigm at Layer 3 (e.g., the ITU-T X.25 protocol), but many rely solely on the Layer 4 transport protocol to detect the few errors that occur end-to-end at this layer. Following a discussion of the ISO layers, we present examples of today's most popular protocol suites.

ISO Reference Model for OSI

Figure 7.5 illustrates the ISO Reference Model for OSI, the objective of which is to solve the problem of heterogeneous DTE and data network communications. However, the OSI model is not a product blueprint. Two companies can therefore build computers consistent with the model, but unable to exchange information. The model is only a framework—meant to be implemented with standards developed for each layer.

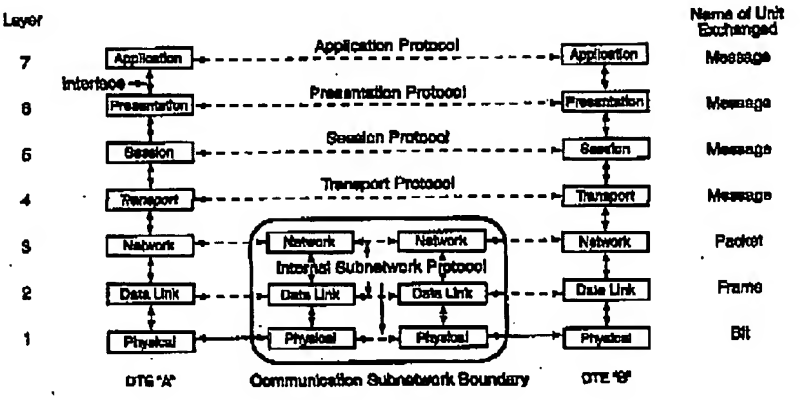


Figure 7.5
ISO reference model for open systems interconnection.

Standards must define services provided by each layer, as well as the protocols between layers. Standards do not dictate how the functions and services are implemented in either hardware or software, so these may differ from product to product.

The International Telecommunication Union (ITU) is a UN treaty organization that considers all technical, operational, and tariff matters for telecommunications worldwide. Its telecommunication standardization committee (ITU-T, formerly the International Consultative Committee for Telegraph and Telephone, or CCITT) functions as the international standards body for the industry.

The results of the ITU-T work are published every four years (following a plenary meeting) as "recommendations" in a series of books commonly referred to by the color of their covers (such as "orange book"). ITU-T recommendations are denoted by A.n, where A is a letter representing a series of recommendations (e.g., V for analog networks, X for digital networks), and n is an identifying number. (See Appendix A for more standards-setting information.)

In Figure 7.5 the protocol stacks to the left and right represent two DTEs connected by a communications subnetwork, shown in the middle. The names and numerical assignments for the seven layers are shown on the left. A summary description of the services specified for each layer follows.

Layer 1: Physical

This layer provides mechanical, electrical, functional, and procedural characteristics to activate, maintain, and deactivate connections for the transmission of unstructured bitstreams over a physical link. The physical link can be connectors and wiring between the DTE and a DCE at a network access point, and fiber optic cable within a network. The ITU X.25 Recommendation defines the interface between *data terminal equipment* (DTE) and *data circuit terminating equipment* (DCE) for terminals operating in the packet mode over public data networks. DCE is a generic term for network-embedded devices that provide attachment points for user devices. Layer 1 involves such parameters as signal levels and bit duration. In the U.S., the RS-232 C standard is commonly used at Layer 1, and bits are the data units exchanged.

Layer 2: Data Link

The Data-Link Layer provides for reliable transfer of data across the physical link. It provides for mapping data units from the next higher (network) layer to frames of data for transmission. Figure 7.1 presents the format for a typical data link frame. The addresses used at Layer 2 are known as *media access control* (MAC) addresses. The data link provides necessary synchronization, error control, and flow control functions. *Link Access Protocol-B* (LAP-B) is an option for Layer 2 in the ITU-T X series recommendations. It is a subset of the ISO-developed *high-level data link control* (HDLC) protocols. In many modern systems, the error-detection function normally associated with connection-oriented operation is performed at Layer 2, but frames with errors are merely discarded. The connection-oriented protocols at higher layers recognize frames not received and request retransmissions at the higher layer.

Layer 3: Network

Layer 3 provides higher-level layers with independence from routing and switching associated with establishing a network connection. Functions include addressing, end-point identification, and service selection when different services are available. Examples of Level 3 protocols are the ITU-T X.25 recommendation and the Internet's IP protocol.

While Layers 1 and 2 can be described as local DTE (station) to DCE (network node) protocols, most of the Layer 3 dialogue is between stations and between nodes. For example, stations address packets to nodes for delivery through the network. There is also, however, a station-to-station aspect of Layer 3 protocols. Stations must provide networks with addressing and other information to route data to other stations.

The network devices that process Layer 3 protocols are referred to as routers. Routers perform the following steps on packets:

- Remove Layer 2 headers
- Check incoming packets for corruption
- Examine packet age and discard packets kept in the network too long
- Filter packets, as required, based on information in the packet
- Determine routes to destinations
- Build new Layer 2 headers
- Forward packets on appropriate output links.

Figure 7.6 illustrates how network DCEs can present a common Layer 3 protocol (in this case, X.25) to attached DTEs and still support different link protocols. This figure also shows how DCE Layer 1 local media connections on the network side (such as copper wire) can be interfaced with long-distance media (such as fiber optic cable). These conversions, together with the entire internal subnetwork operation are accomplished transparently to user DTEs, which are presented with an X.25 interface.

Layer 4: Transport

In conjunction with the underlying Network, Data-Link, and Physical Layers, the Transport Layer provides end-to-end (station-to-station) control of transmitted data and optimizes use of network resources. This layer exists to provide transparent data transfer between Layer 5 session entities. In ISO terminology, an entity is the network processing capability (hardware, software, or both) that implements functions in a particular layer. Thus, entities are identified for each layer, e.g., the Layer 5 session.

Transport Layer services are provided to upper layers in order to establish, maintain, and release transparent data connections over two-way, simultaneous data transmission paths between pairs of transport addresses. The transport protocol capabilities needed depend upon the quality of the underlying layer services.

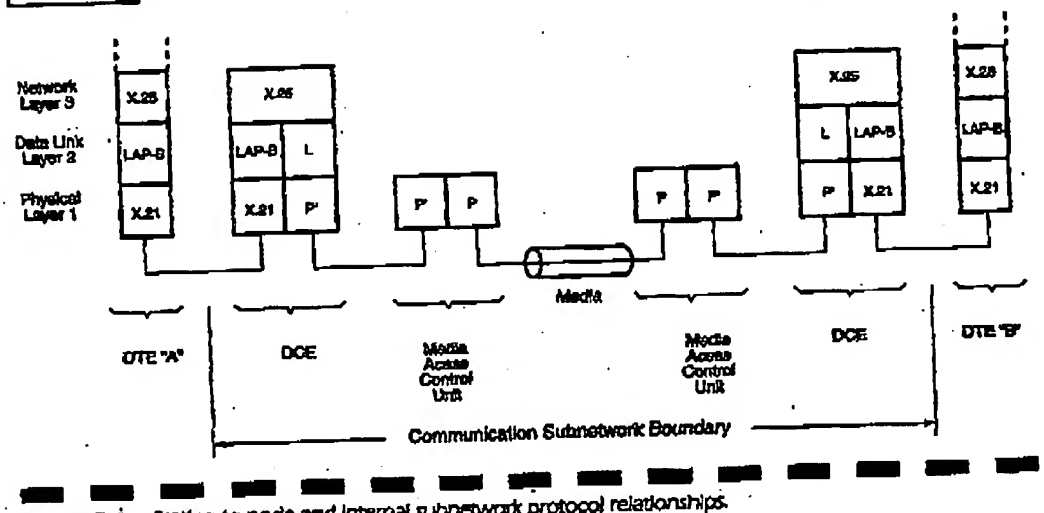


Figure 7.6 Station-to-node and internal subnetwork protocol relationships.

With reliable, error-free virtual circuit network service, a minimal Transport Layer is required. If the lower layers provide connectionless service, then the transport protocol must implement error detection and recovery, and other functions. ISO has defined five transport protocol classes, each consistent with a different underlying service.

Layer 5: Session

A *session* is a connection between stations that allows them to communicate. For example, a host processor may need to establish multiple sessions simultaneously with remote terminals to accomplish file transfers with each.

The purpose of the Session Layer is to enable two presentation entities at remote stations to establish and use transport connections by organizing and synchronizing their dialogue and managing the data exchange. Very few protocol implementations today include a separate Session Layer; the functions of the Session Layer are combined in Layer 7, the Application Layer protocol.

Layer 6: Presentation

The Presentation Layer delivers information to communicating application entities in a way that preserves meaning while resolving syntax

differences. Toward this objective, Layer 6 can provide data transformation (e.g., data compression or encryption), formatting, and syntax selection.

Virtual terminal protocol, a Layer 6 protocol, hides differences in remote terminals from application entities by making the terminals all appear as generic or virtual terminals. When two remote host processors use virtual-terminal protocols, terminals appear as locally attached to either host. Like the Session Layer, the Presentation Layer is combined in the Application Layer protocol in most modern implementations.

Layer 7: Application

The Application Layer enables a computer's application process to access the OSI environment. It serves as the passageway between application processes using open systems interconnection to exchange information. All services directly usable by the application process are provided by this layer. Services include identification of intended communications partners, determination of the current availability of the intended partners, establishment of the authority to communicate, agreement on responsibility for error recovery, and agreement on procedures to maintain data integrity.

TCP/IP Protocols

In Chapter 2, the creation of the packet-switched network concept by the Advanced Research Projects Agency was chronicled. At the heart of the rapid growth of ARPA's original network, culminating in today's ubiquitous Internet, was a set of simple, yet powerful protocols that have stood the test of time to become the most popular solutions for internetworking ever devised. Although these protocols predated OSI reference model protocols discussed above, they implement equivalent procedures but in four layers as opposed to seven.

Figure 7.7 compares OSI and ARPA protocol models. The heart of ARPA's model is the Internet Layer, which manages message flow between host computers and intermediate packet switches. The protocol developed for this layer is known as the *Internet Protocol*, the "IP" in TCP/IP. IP is designed with the assumption that underlying communication subnetworks provide perfect communication channels

(i.e., it is a connectionless protocol). ARPA's mission to support military networks dictated a reliable connection for the *Host-to-Host* Layer; ARPA responded by specifying *Transmission Control Protocol* (TCP) for applications requiring reliable end-to-end service.

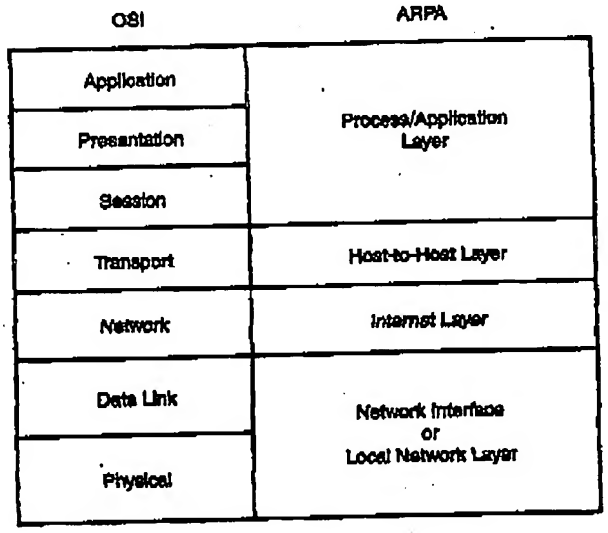


Figure 7.7 OSI and ARPA protocol models.

With TCP and IP providing underpinnings, a series of Process/Application protocols were developed to perform the real work for users. These include *Telecommunications Network* (TELNET) protocol to allow remote host access and terminal emulation, *File Transfer Protocol* (FTP) to transfer files between two host systems, *Simple Mail Transfer Protocol* (SMTP) to send electronic mail (e-mail) messages from one host to another, and the *Simple Network Management Protocol* (SNMP) to enable central management of network resources. The key to Internet growth is the widespread adoption of simple, efficient protocols that can be used across many computer platforms. This feat was and is accomplished not by any controlling government authority, but by a largely volunteer community operating under a self-governing structure designed to promote maximum user community participation and unbiased consideration of ideas.

At the center of this structure is the Internet Society and one of its components, the 13-member Internet Architecture Board (IAB). One of

its task forces, the Internet Engineering Task Force (IETF), coordinates the technical aspects of the Internet and its protocols. The IETF produces numerous protocol standards, known as *Request for Comments* (RFC) documents. To become an Internet standard, a proposal undergoes several levels of testing and revision and is finally adopted by the IETF through a democratic voting process. Only after significant implementation and operational experience can a Draft Standard be elevated to an Internet Standard. Figure 7.8 depicts relationships among TCP/IP protocols and names the RFC reference for each standard.

Figure 7.8 Internet protocol summary.

OSI	Protocol Implementation						ARPA
Application	File Transfer	Electronic Mail	Terminal Emulation	File Transfer	Client/ Server	Network Management	Process/ Application Layer
Presentation	File Transfer Protocol (FTP) RFC 959	Simple Mail Transfer Protocol (SMTP) RFC 821	TELNET Protocol RFC 821	Trivial File Transfer Protocol (TFTP) RFC 821	Network File System Protocol (NFS) RFCs 1014, 1057, 1094	Simple Network Management Protocol (SNMP) RFC 821	
Session							
Transport	Transmission Control Protocol (TCP) RFC 793			User Datagram Protocol (UDP) RFC 768			Host-to-Host Layer
Network	Address Resolution Protocol (ARP) RFC 826		Internet Protocol (IP) RFC 791	Internet Control Message Protocol (ICMP) RFC 792			Internet Layer
Data Link	Network Interface Cards: Ethernet, Token Ring, ARCNET, WAN RFCs 894, 1042, 1201						Network Interface or Local Network Layer
Physical	Transmission Media: Twisted Pair, Coax, Fiber Optics, Wireless, etc.						

The following sections describe in detail the information contained in the headers for the IP Layer (as an example of a connectionless protocol) and the TCP Layer (a connection-oriented protocol). These examples provide insight into the workings of protocols in general as well as details of specific protocols themselves.

Internet Protocol

IP was developed as a connectionless protocol at what we now refer to as Layer 3. As such, it is primarily concerned with delivering packages of bits from sources to destinations over interconnected systems of networks. As in all protocol layers, the "package of bits" includes original user information and any header and trailer bits added by higher-layer protocols. Header bits are control bits added to the beginning of

a package for use by receiving protocol processors at the corresponding layer. Trailer bits are added at the end of the package for the same purpose and may or may not be used in any given protocol. IP uses only header bits to perform its functions. Figure 7.9 illustrates the structure of an IP packet.

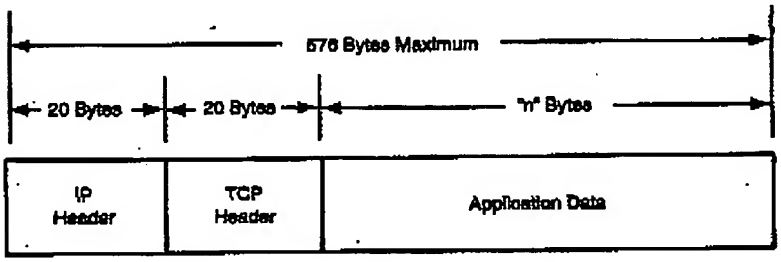


Figure 7.9
Structure of an IP packet.

The major functions that must be dealt with by IP headers are addressing and fragmentation. *Addressing* is obviously needed to route packets to destinations, but *what is fragmentation* and *why is it necessary*? The local and long distance networks that IP packets must traverse may have different Layer 2 frame sizes. The complete IP packet, including the Layer 4 package plus the IP header, must exactly fit into these frames. If Layer 4 packages are shorter than the required length, they can be padded with null bytes. However, if too long, they must be broken into several pieces (i.e., fragments) that will fit. The IP header uses *fields* to help receivers reassemble fragments into original package formats.

Figure 7.10 identifies the specific fields within an IP header, using a standardized format for displaying protocol fields. Each horizontal group of bits (called a *word*) is 32 bits wide. The order in which bits are actually transmitted is from left to right and top to bottom. Note that the minimum header length is five words, or 20 bytes. The first word contains fields for IP version, header length, type of service, and total packet length. The second word comprises three fields supporting fragmentation and reassembly: a fragment identifier, a set of flags indicating whether a packet is the last fragment, and an offset to indicate where a fragment belongs in the complete message. The next word contains time to live which is decreased each time a packet passes through a router. When the TTL value reaches 0, the packet is destroyed. This prevents misaddressed packets from being routed forever. A protocol field identifies the higher-level protocol in use (e.g., TCP).

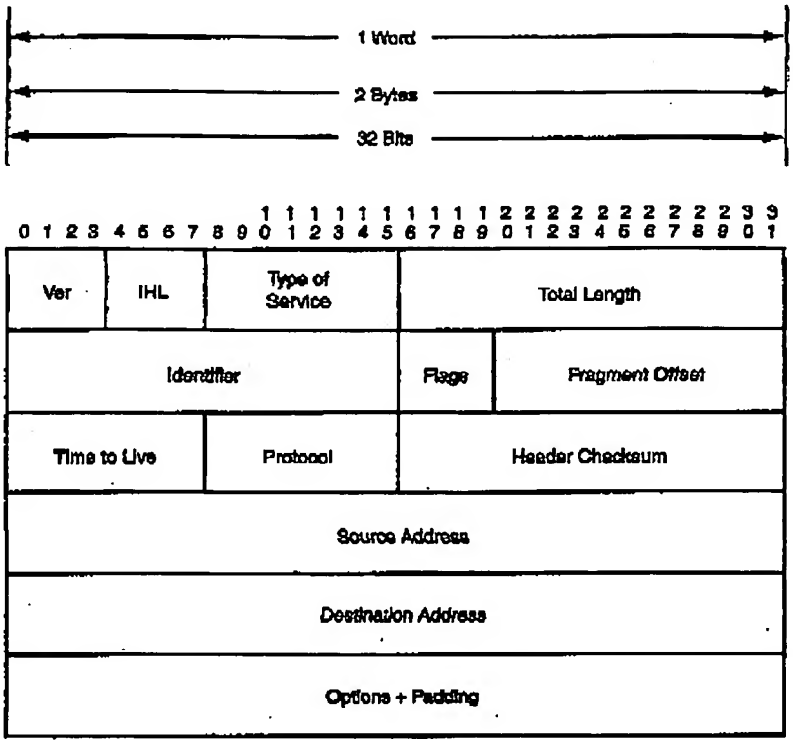


Figure 7.10 Internet Protocol header format.

The fourth and fifth words contain 32-bit source and destination addresses, respectively. The destination address is used for routing; the source address can be used for security screening and filtering or other processing at destinations. Addresses are normally represented in dotted decimal notation, in which each byte is assigned a decimal number from 0 to 255 (e.g., 150.200.100.5). Each IP address is divided into network ID and host ID parts. A central authority assigns network IDs and local network administrators assign host IDs. Routers send packets to a network based on its network ID and that network completes the delivery to the host. The number of bits in the address assigned to a network ID depends on the size of that network.

It should be noted that protocol headers add *overhead* to information bytes carried by a network (overhead not present in circuit switched data networks). The packet size of Internet IP packets is 576

bytes, of which at least 20 bytes (35 percent) is IP overhead. The addition of Layer 4 and Layer 2 overhead bytes typically raises the packet overhead penalty to 8 percent.

Transmission Control Protocol

As stated previously, IP does not guarantee reliable packet delivery. This function falls to the Layer 4 Transmission Control Protocol (TCP). In fact, TCP handles six functions: basic data transfer, reliability, flow control, multiplexing, connections, and precedence/security. Fields in TCP headers are shown in Figure 7.11. Headers include a sequence number used to ensure that data packets arrive in sequence, one requirement of reliable transmission. An acknowledgment number field verifies data receipt. TCP is a *Positive Acknowledgment with Retransmission* (PAR) protocol. When data are received correctly with expected sequence numbers, acknowledgment numbers are sent back to senders. If transmitting stations do not receive proper acknowledgements within specified times, they retransmit. No negative acknowledgments are sent.

Flow control is implemented using a header window field. Along with acknowledgment numbers, TCP segment receivers send *window size* data back to transmitters. *Window size* is the number of bytes of data receivers can accept and store in their buffers before sending acknowledgments. Small window size necessitates large numbers of acknowledgment transmissions, consuming bandwidth that could otherwise be allocated to user data. Large window sizes necessitate correspondingly large host buffers, a hardware penalty. Window size is determined during TCP connection setup procedures and can be changed by hosts during sessions as conditions change. Referred to as "sliding window operation" this capability enables flow rate control among hosts.

Two other fields in the TCP header specify source and destination port numbers. These port numbers correspond to specific end user processes (i.e., applications implemented by upper-layer protocols). The combination of a port number and an Internet address is called a *socket*. Since a given host can be a multi-function system (i.e., support several applications simultaneously), more than one socket can be active in a host. A TCP connection is the association of a pair of sockets in two machines. TCP provides true multiplexing of data connections through this mechanism.

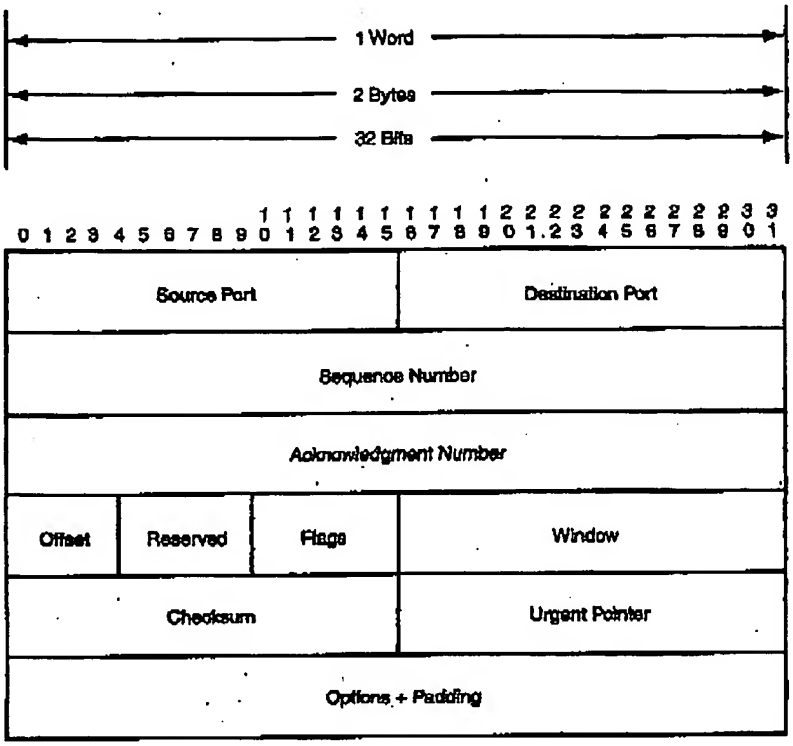


Figure 7.11 Transmission Control Protocol header format.

Other Protocols

Besides the application protocols mentioned above, other TCP/IP protocols exist at Layers 3 and 4. The *User Datagram Protocol* (UDP) provides connectionless transport services for applications not requiring TCP reliability. Its shorter header and lack of connection setup overhead make it more efficient when the amount of data to be transmitted is small. Layer 3 protocols exist to perform address translations necessary to deliver packets to specific hardware addresses (e.g., the Address Resolution Protocol and the Dynamic Host Configuration Protocol) and to communicate "network health" status among hosts (the Internet Control Message Protocol). A detailed discussion of all TCP/IP mechanisms is beyond this book's scope. Readers needing more details can consult the library of RFCs available on the Internet at **http://www.faqs.org/rfcs**.

New Directions

TCP/IP's greatest strength lies in its planner's (the IETF) flexible, distributed, and highly dynamic methods for implementing changes and adapting to market forces. The planners, vendors and manufacturers and Internet users have resisted numerous attempts to replace TCP/IP with legislated "standard" protocol suites (most notably OSI protocols). The current driving force for TCP/IP evolution is the growth of the commercial Internet with its virtually unlimited potential and its users' near insatiable appetite for innovative applications.

Two major factors are causing stress for the Internet today: its sheer size and the emergence of real-time streaming applications. First, the number of connected computers on the Internet is rapidly exhausting the currently defined IP address space. The four-byte address limitation and the way addresses are assigned to networks cannot support the current growth rate. For several years there has been pressure on the IETF to expand the address format. Second, streaming applications such as video, music, and voice are significantly increasing Internet traffic and the demand for low latency (i.e., low delay) that cannot be guaranteed by the current connectionless IP routing paradigm.

Latency across a network is just one of several performance parameters used to define Quality of Service (QoS) for a network. Different applications may require different QoS specifications. For example, users tolerate high error rates in voice traffic, but not excessive delay or varying delays. In contrast, electronic mail recipients won't tolerate message content errors but are relatively insensitive to reasonable fixed or variable delays. The Internet's current inability to support multiple QoS levels, in accordance with different application requirements, is a major shortcoming.

Solutions to the above problems come in two arenas. First, changes in the use of fields in the IP header to implement the means for supporting multiple QoS levels in the Internet backbone (some suitable for video, music, and voice applications) are being adopted by the IETF. Second, a new version of IP, known for historical reasons as IP Version 6 (IPv6), has been adopted to address long-term resolution of both address and QoS issues.

DIFFSERV AND MPLS

The IETF is now examining two standards to help solve the IP QoS problem: Differentiated Services (*Diffserv*) and Multiprotocol Label Switching (MPLS). These techniques address the problem in very differ-

ent ways and have different implications on the Internet's architecture. While each can exist without the other, they can be used together.

Diffserv is a Layer 3 solution and uses IP's type-of-service field to carry information about IP packet service requirements. It relies on traffic conditioners at the edge (boundary) of networks to indicate each packet's requirements based on the needs of the application. For example, packets marked with an *expedited forwarding* (EF) indication receive better processing during the forwarding process than normal packets. This may include assignment to special priority queues. Of course, as the standard evolves, Diffserv-capable routers will have to be installed in the Internet infrastructure. One advantage of Diffserv is that router processing decisions are made on a per-packet basis, not on a per-session basis, allowing more flexibility for ISPs to configure routing algorithms.

Diffserv will be the first of these QoS mechanisms to be ratified as a standard. Since it specifies QoS at Layer 3, it will be implemented at the edge of the network in user devices, and be transported over any Layer 2 infrastructure. As an example, Microsoft is going to include Diffserv capabilities in its upcoming release of the Windows 2000 operating system.

MPLS, in contrast to Diffserv, maps Layer 3 traffic to connection-oriented Layer 2 transports. It adds a label containing specific routing information to each IP packet and allows routers to assign explicit paths to various classes of traffic. MPLS requires investing in sophisticated label-switching routers capable of reading new header information and assigning packets to specific paths. As such, it will likely be implemented at the core of carrier networks and may receive QoS packet requirement information from Diffserv fields.

Routing efficiency is obtained in networks by relieving each router in the path of the burden of running its own network-layer routing algorithm. In this alternative the routing path is calculated only once and encapsulated in a label, an extra 32 bits added to the front of current IP headers. Subsequent routers read the label and follow the path instructions. The path calculation done initially may depend on packet QoS requirements. Finally, since MPLS specifies complete paths for streams of packets, it can easily map such streams onto connection-oriented Layer 2 paths.

IPv6

The traditional IP protocol described above carries version number four (the designation IPv4 is used when it is necessary to distinguish it

from the new version six). The IETF began investigating options to address its shortcomings in 1990 and published its recommendations in January 1995. The revision was assigned version number six (an experimental protocol had been assigned version number five, but was never deployed) and is commonly known as IPv6. Of course, changes to IP cannot be done in isolation and at least 60 current TCP/IP standards must be revised to accommodate IPv6. Although many changes were and are being made, the most important for users are the expansion of the addresses and the inclusion of a flow label in headers.

Flows are defined as streams of packets associated with a particular application. As discussed above, the identification of flows and flow characteristics is an important part of implementing different QoS levels in the Internet. While IPv6 does not specify how flow labels in headers are to be used, it provides capabilities for source devices and routers to identify and process specific flows. As IPv6 is adopted, the Internet community will use these capabilities to implement what is necessary to support ever-evolving applications.

The major change in IPv6 is implementation of 128-bit addresses to replace current 32-bit addresses. Obviously, the four-part dotted decimal notation used for current IP addresses is no longer applicable. The preferred representation is:

```
x:x:x:x:x:x:x:x
```

where each "x" represents 16 bits. The 16 bits in each address part are represented using four hexadecimal digits (i.e., 0—9, A, B, C, D, E, F representing values from 0 to 15). For example, an IPv6 address could be:

```
FEDC:BA98:4387:3298:EFDA:AB65:4523:853A
```

Leading zeros are not required in representations for any address part. In addition, if long strings of zeros appear in the address (i.e., 0:0:0), a double colon "::" may be used to indicate multiple groups of 16 bits of zeros. The use of the double colon is restricted to one application in an address. Two examples of this address simplification are:

```
1080:0:0:0:8:800:200C:417A -> 1080::8:800:200C:417A
0:0:0:0:0:0:0:1 -> ::1
```

Different options are still being considered for implementing hierarchies within address spaces, similar to the network-host hierarchy in IPv4. Many of these options include using a 48-bit network interface

ID, unique to each hardware interface card, as the lower 48 bits of the new IP address. In addition, special types of address formats have been defined that deal with address problems encountered while making the transition from IPv4 to IPv6. These formats are used at boundaries between IPv4 and IPv6 networks and use existing IPv4 node 32-bit addresses as the lower 32 bits of an IPv6 address for reference within the IPv6 network.

Note that expanding address bytes adds more overhead in TCP/IP systems. Retiring some fields and using optional extension headers for other functions has held the minimum size of the new IPv6 header to 40 bytes (vs. 20 bytes for the IPv4 header). Overall overhead penalties discussed previously are now 12 percent.

TRANSITION TO IPV6

Clearly, IPv6 developers did not envision upgrading the Internet to IPv6 all at once. With millions of connected devices and exponential growth, the transition of the Internet to IPv6 represents the most ambitious undertaking of its kind in history. Since the Internet is made up of diverse systems from many manufacturers, it is expected that many systems may not be upgraded for years, if at all. Therefore, strategies have been defined to allow IPv4 and IPv6 networks and devices to coexist.

Two mechanisms have been proposed to accomplish this function: a dual IP layer, and IPv6-over-IPv4 tunneling. The dual-layer approach is the simplest and calls for both protocols to be implemented on new or upgraded devices. Such devices can then communicate to IPv6 devices using IPv6 and IPv4 devices using IPv4. Conversion capabilities make this kind of device applicable to gateway functions between network types.

Tunneling is an approach in which entire IPv6 packets are encapsulated inside IPv4 packets (i.e., an IPv4 header is put on top of an entire IPv6 packet, including the IPv6 header). This allows resulting packets to be routed through existing IPv4 networks. At the end of the "tunnel," dual-mode devices remove IPv4 headers and processes IPv6 packets.

For either scenario to operate, networks must provide information about both types of addresses, the configuration and addresses of gateways, and tunnel endpoints. The development of this infrastructure in a system as large as the Internet is an extremely difficult task. Hence, the first applications of IPv6 will probably be within isolated subnetworks with IPv4 used for transport over a wide area. When IPv6 becomes more widely implemented in commercial software, the transition will start to take place. As with all attempts to change TCP/IP and the Internet, the pace of change will be driven by a best-